

Once Gödel dreamt about codes. And he foresaw a very interesting puzzle involving finite sequences of bits called *h*-sequences. An *h*-sequence is a string defined recursively as follows:

$$\begin{aligned}\langle hseq \rangle & ::= 0 \\ \langle hseq \rangle & ::= 1\langle hseq \rangle\langle hseq \rangle\end{aligned}$$

For instance, 0, 11000, and 101100100 are *h*-sequences but 1, 11001, and 111100100 are not.

Gödel's dream was of a tall order of *h*-sequence fun because the strings had incomplete information. More precisely, strings did not only contain bits '0' and '1', but they also contained the question mark symbol '?'. The occurrence of a question mark symbol in a string indicates that any bit can go in such a position of the string. For example, the sequence 101?0 can actually have the *h*-sequence 10100 and the string 10110 (which is not an *h*-sequence) as instances.

Given a string made of bits, possibly containing question mark symbols, your task is to write a program that computes the maximum number of *h*-sequences that concatenated together result in an instance of such a string.

## Input

The input consists of several test cases, each one defined by a line containing a string *s* made of characters '0', '1', and '?', whose length is between 1 and  $10^4$  inclusive.

## Output

For each test case, output a line with the maximum number of *h*-sequences that concatenated together result in an instance of *s*.

## Sample Input

```
0
10100
??1?
?1??
1?010100
???1????
```

## Sample Output

```
1
1
0
2
2
6
```